| | | | | | |
|---|---|---|---|---|---|
| Universidade Federal de São João del-Rei | **COORDENADORIA DO CURSO DE ENGENHARIA MECATRÔNICA**<br>**PLANO DE ENSINO** | | | | |
| **Disciplina:** Programação Orientada a Objetos | | | **Período:** 3° | | **Currículo:** 2010 |
| **Docente Responsável: Samuel M. A. Araújo** | | | **Unidade Acadêmica:** DTECH | | |
| **Pré-requisito:** Algoritmos e Estrutura de Dados I | | | **Correquisito:** Não há | | |
| **C.H. Total:** 72 | **C.H. Prática:** 18 | **C.H. Teórica:** 54 | **Grau:** **Bacharelado** | **Ano:** 2023 | **Semestre:** 2º |

## EMENTA

Filosofia e Princípios. Qualidade de Software. Modularidade. Reutilização de Software. Abstração. Tipos Abstratos de Dados. Objetos. Classes. Atributos. Métodos. Encapsulamento. Herança. Hierarquia de Classes. Interface. Polimorfismo. Análise e Projeto Orientados por Objetos. Tratamento de Exceção. Genericidade. Parametrização de Classes. Linguagens Orientadas por Objetos: Java, C++, entre outras. Aulas práticas em laboratório.

## OBJETIVOS

Introduzir os conceitos fundamentais do paradigma de orientação a objetos e a sua aplicação à programação orientada a objetos. Apresentação de conceitos de linguagens de programação orientadas a objetos.

## CONTEÚDO PROGRAMÁTICO

1 Visão geral
- 1.1 Classes e objetos
- 1.2 Modelagem orientada a objetos
- 1.3 Modularização
- 1.4 Reutilização de código

2 Classes e objetos
- 2.1 Declaração de classes
- 2.2 Criação de objetos
- 2.3 Atributos e métodos
- 2.4 Construtores
- 2.5 Membros *static*
- 2.6 Vetor de objetos

3 Herança
- 3.1 Definição
- 3.2 Hierarquia de classes
- 3.3 O ponteiro *this*
- 3.3 Sobrecarga de métodos
- 3.4 Herança múltipla
- 3.5 Conversão entre tipos

4 Encapsulamento
- 4.1 Definição
- 4.2 Interface de acesso de uma classe
- 4.3 Membros *public*

4.4 Membros *private*
4.5 Membro *protected*
4.6 Sobrecarga de métodos na hierarquia de classes

5 Polimorfismo
5.1 Definição
5.2 Classes abstratas e interfaces
5.3 Métodos abstratos
5.4 Classes e funções amigas

6 Modelagem orientada a objetos
6.1 Introdução a UML
6.2 Processo de desenvolvimento de programas
6.2 Diagrama de caso de uso
6.3 Diagrama de sequência
6.4 Diagrama de classes

7 Tratamento de Exceção
7.1 Definição
7.2 Tipos
7.3 Utilização

## METODOLOGIA DE ENSINO

- Aulas expositivas com diálogos e atividades iterativas.
- Aprendizagem através de solução de problemas, de forma que o aluno aprenda com a prática.
- Realização de práticas investigativas, em laboratório, com o uso de uma linguagem de programação.
- Para cada item do conteúdo programático será dada uma atividade, a qual será apresentada pelo aluno para o professor.
- Exercícios extraclasse, provas e trabalhos práticos individuais e em grupos, para aprendizado aprofundado dos conceitos e técnicas estudadas.

## CONTROLE DE FREQUÊNCIA E CRITÉRIOS DE AVALIAÇÃO

Cem pontos distribuídos ao longo do semestre da seguinte maneira:
- 3 Provas - cada uma valendo 20 pontos (total 60 pontos)
- Listas de exercícios e práticas de laboratório ao longo do período - 20 pontos
- Trabalho Prático – 20 pontos

Prova Substitutiva: o aluno que ficar abaixo da média de 60% ao final do semestre, ou vier a perder alguma aplicação de prova, poderá submeter-se à uma prova de substituição/reposição no valor de 20 pontos. Neste caso, a nota da prova substitutiva substituirá a nota da menor prova realizada pelo aluno, ou irá repor a nota da prova perdida. Essa substitutiva abordará todo o conteúdo da disciplina. Ao final do semestre, o aluno que não atingir 60 pontos totais, não será aprovado.

Será feito o controle de presença em todas as aulas. Por tratar-se de um curso presencial, o comparecimento do corpo discente às aulas é obrigatório. Em nenhuma hipótese será concedido abono de falta, exceto nos casos previstos na legislação em vigor e no estatuto da universidade. O discente que não comparecer a 75% das aulas será reprovado por infrequência.

| BIBLIOGRAFIA BÁSICA |
|---|
| 1. MIZRAHI, V. V. Treinamento em Linguagem C++ - Módulo 2. 2ª Edição, Makron Books, 2006.<br>2. DEITEL, H. M.; DEITEL, P. J. C++ - Como Programar, 3ª Edição, Bookman, 2001<br>3. DEITEL, H. M.; DEITEL, P. J. Java - Como Programar, 6ª Edição, Bookman, 2005. |
| **BIBLIOGRAFIA COMPLEMENTAR** |
| 1. STROUSTRUP B. A Linguagem de Programação C++, 3ª Edição, Bookman,1999.<br>2. BOOCH, G. Object-oriented analysis and design with Applications, 2ª Edição, Addison Wesley, 1994.<br>3. COX, B. J. Object-Oriented Programming, Addison Wesley, 1986. (Nacional: Editora Makron 1991).<br>4. SATIR G.; BROWN D.: C++: The Core Language, O'Reilly, 1995.<br>5. STROUSTRUP B.: An Overview of the C++ Programming language. Handbook of Object Technology. CRC Press, 1998. |

| | |
|---|---|
| | Aprovado pelo Colegiado em / / |
| Docente Responsável | Prof. Edgar Campos Furtado<br>Coordenador do Curso de Engenharia Mecatrônica |

**MINISTÉRIO DA EDUCAÇÃO**
**UNIVERSIDADE FEDERAL DE SÃO JOÃO DEL-REI**
**SISTEMA INTEGRADO DE PATRIMÔNIO, ADMINISTRAÇÃO E CONTRATOS**

**FOLHA DE ASSINATURAS**

*Emitido em 18/08/2023*

**PLANO DE ENSINO Nº PE POO 2023/2/2023 - CEMEC (12.56)**
**(Nº do Documento: 3045)**

**(Nº do Protocolo: 23122.032084/2023-64)**

***(Assinado digitalmente em 21/08/2023 08:57 )***
EDGAR CAMPOS FURTADO
*COORDENADOR DE CURSO*
*CEMEC (12.56)*
*Matrícula: ###424#4*

***(Assinado digitalmente em 19/08/2023 09:44 )***
SAMUEL MOREIRA ABREU ARAUJO
*PROFESSOR DO MAGISTERIO SUPERIOR*
*DTECH (12.27)*
*Matrícula: ###282#8*

Visualize o documento original em https://sipac.ufsj.edu.br/public/documentos/ informando seu número: **3045**, ano: **2023**, tipo: **PLANO DE ENSINO**, data de emissão: **18/08/2023** e o código de verificação: **5156bf692a**